DEPARTMENT OF
DATA SCIENCE

DS211
Linear and Nonlinear Models

Graphs for Poster

Graph god

Setup

```
# ----------------------------------------------------
# Set working directory
# ----------------------------------------------------
setwd("C:/Users/PC/Documents/2ND YEAR 1ST SEMESTER/L&NL/FINALS/DATAS")

# Load required packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.2

## Warning: package 'ggplot2' was built under R version 4.5.2

## Warning: package 'tibble' was built under R version 4.5.2

## Warning: package 'tidyr' was built under R version 4.5.2

## Warning: package 'readr' was built under R version 4.5.2

## Warning: package 'purrr' was built under R version 4.5.2
```

```
## Warning: package 'dplyr' was built under R version 4.5.2

## Warning: package 'stringr' was built under R version 4.5.2

## Warning: package 'forcats' was built under R version 4.5.2

## Warning: package 'lubridate' was built under R version 4.5.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# ------------------------------------------------
# Helper function: read exactly ONE CSV per folder
# ------------------------------------------------
read_single_csv <- function(folder_path) {
  csv_file <- list.files(folder_path, pattern = "\\.csv$", full.names = TRUE)

  if (length(csv_file) != 1) {
    stop(paste("Expected 1 CSV file in", folder_path,
               "but found", length(csv_file)))
  }

  read_csv(csv_file)
}

# ------------------------------------------------
# Read datasets from DATAS folders
# ------------------------------------------------
birth_rate <- read_single_csv(
  file.path(getwd(), "crude-birth-rate")
)
```

```
## Rows: 18722 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Birth rate - Sex: all - Age: all - Variant: estimates
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
age_dependency <- read_single_csv(
  file.path(getwd(), "age-dependency-ratio-of-working-age-population")
)
```

```
## Rows: 18944 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Total dependency ratio - Sex: all - Variant: estimates
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
child_mortality <- read_single_csv(
  file.path(getwd(), "child-mortality-igme")
)
```

```
## Rows: 13700 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Child mortality rate of children aged under five years, per 1...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
crude_death_rate <- read_single_csv(
  file.path(getwd(), "crude-death-rate")
)
```

```
## Rows: 18722 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Death rate - Sex: all - Age: all - Variant: estimates
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gdp_per_capita <- read_single_csv(
  file.path(getwd(), "gdp-per-capita-penn-world-table")
)
```

```
## Rows: 10907 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, GDP per capita (output, multiple price benchmarks)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gov_revenues <- read_single_csv(
  file.path(getwd(), "government-revenues-as-a-share-of-gdp-imf")
)
```

```
## Rows: 3340 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, 17.1.1 - Total government revenue as a proportion of GDP (%) ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gov_spending <- read_single_csv(
  file.path(getwd(), "historical-gov-spending-gdp")
)
```

```
## Rows: 9242 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Entity, Code, 1025229-annotations
## dbl (2): Year, Government expenditure (% of GDP)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
avg_age_mothers <- read_single_csv(
  file.path(getwd(), "period-average-age-of-mothers")
)
```

```
## Rows: 18958 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Mean age at childbearing, historical
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# ---------------------------------------------------
# Standardize variable names (MATCHING ACTUAL COLUMNS)
# ---------------------------------------------------

birth_rate <- birth_rate %>%
  select(
    country = Entity,
    year = Year,
    birth_rate = `Birth rate - Sex: all - Age: all - Variant: estimates`
  )

age_dependency <- age_dependency %>%
  select(
    country = Entity,
    year = Year,
    age_dependency_ratio = `Total dependency ratio - Sex: all - Variant: estimates`
  )
```

```r
child_mortality <- child_mortality %>%
  select(
    country = Entity,
    year = Year,
    child_mortality_rate =
      `Child mortality rate of children aged under five years, per 100 live births`
  )

crude_death_rate <- crude_death_rate %>%
  select(
    country = Entity,
    year = Year,
    crude_death_rate = `Death rate - Sex: all - Age: all - Variant: estimates`
  )

gdp_per_capita <- gdp_per_capita %>%
  select(
    country = Entity,
    year = Year,
    gdp_per_capita = `GDP per capita (output, multiple price benchmarks)`
  )

gov_revenues <- gov_revenues %>%
  select(
    country = Entity,
    year = Year,
    gov_revenue_gdp =
      `17.1.1 - Total government revenue as a proportion of GDP (%) - GR_G14_GDP`
  )

gov_spending <- gov_spending %>%
  select(
    country = Entity,
    year = Year,
    gov_spending_gdp = `Government expenditure (% of GDP)`
  )

avg_age_mothers <- avg_age_mothers %>%
  select(
    country = Entity,
    year = Year,
    avg_age_mothers = `Mean age at childbearing, historical`
  )


# ------------------------------------------------
# Merge all datasets into ONE cross-country dataset
# ------------------------------------------------
data_full <- birth_rate %>%
  left_join(age_dependency, by = c("country", "year")) %>%
  left_join(child_mortality, by = c("country", "year")) %>%
  left_join(crude_death_rate, by = c("country", "year")) %>%
  left_join(gdp_per_capita, by = c("country", "year")) %>%
```

```r
  left_join(gov_revenues, by = c("country", "year")) %>%
  left_join(gov_spending, by = c("country", "year")) %>%
  left_join(avg_age_mothers, by = c("country", "year"))

# ------------------------------------------------
# Filter data for CROSS-COUNTRY ANALYSIS in 2023
# ------------------------------------------------
data_2023 <- data_full %>%
  filter(year == 2023)

# ------------------------------------------------
# Remove observations with missing values
# (Final regression-ready dataset)
# ------------------------------------------------
data_2023_clean <- data_2023 %>%
  drop_na(
    birth_rate,
    avg_age_mothers,
    age_dependency_ratio,
    child_mortality_rate,
    crude_death_rate,
    gdp_per_capita,
    gov_revenue_gdp,
    gov_spending_gdp
  )

# ------------------------------------------------
# Data checks
# ------------------------------------------------
glimpse(data_2023_clean)
```

```
## Rows: 65
## Columns: 10
## $ country              <chr> "Austria", "Bahamas", "Bosnia and Herzegovina", "~
## $ year                 <dbl> 2023, 2023, 2023, 2023, 2023, 2023, 2023, 2023, 2~
## $ birth_rate           <dbl> 8.439, 10.852, 7.714, 12.322, 9.218, 9.078, 8.963~
## $ age_dependency_ratio <dbl> 52.55727, 42.47221, 53.48485, 44.02432, 57.15226,~
## $ child_mortality_rate <dbl> 0.3126400, 1.2686870, 0.6042108, 1.4441526, 0.607~
## $ crude_death_rate     <dbl> 10.210, 8.713, 13.457, 7.077, 15.153, 7.935, 6.49~
## $ gdp_per_capita       <dbl> 62852.680, 34392.930, 19023.846, 18712.658, 28219~
## $ gov_revenue_gdp      <dbl> 49.42, 19.91, 41.24, 37.90, 37.04, 41.96, 25.10, ~
## $ gov_spending_gdp     <dbl> 51.97942, 24.15769, 42.45777, 45.44920, 37.74231,~
## $ avg_age_mothers      <dbl> 31.167, 28.841, 29.037, 27.813, 27.948, 31.413, 3~
```

```r
summary(data_2023_clean)
```

```
##    country                year        birth_rate     age_dependency_ratio
##  Length:65          Min.   :2023   Min.   : 4.568   Min.   :22.00
##  Class :character   1st Qu.:2023   1st Qu.: 8.919   1st Qu.:48.89
##  Mode  :character   Median :2023   Median :10.799   Median :53.88
##                     Mean   :2023   Mean   :14.673   Mean   :55.17
##                     3rd Qu.:2023   3rd Qu.:18.616   3rd Qu.:58.70
```

```
##                     Max.   :2023   Max.   :35.222   Max.   :85.49
##  child_mortality_rate crude_death_rate gdp_per_capita  gov_revenue_gdp
##  Min.   :0.2106       Min.   : 0.932   Min.   : 2776   Min.   : 5.51
##  1st Qu.:0.3944       1st Qu.: 5.960   1st Qu.:11245   1st Qu.:24.19
##  Median :0.8073       Median : 7.702   Median :26244   Median :33.11
##  Mean   :1.4746       Mean   : 8.152   Mean   :31645   Mean   :32.85
##  3rd Qu.:1.6575       3rd Qu.: 9.783   3rd Qu.:45787   3rd Qu.:41.96
##  Max.   :9.5031       Max.   :15.153   Max.   :98778   Max.   :62.18
##  gov_spending_gdp avg_age_mothers
##  Min.   :10.82    Min.   :25.97
##  1st Qu.:25.72    1st Qu.:28.29
##  Median :33.17    Median :29.22
##  Mean   :34.73    Mean   :29.66
##  3rd Qu.:43.20    3rd Qu.:31.17
##  Max.   :74.42    Max.   :33.42
```

```r
# Missing values check
sum(is.na(data_2023))         # Before cleaning
```

```
## [1] 393
```

```r
sum(is.na(data_2023_clean))   # After cleaning
```

```
## [1] 0
```

Graphs

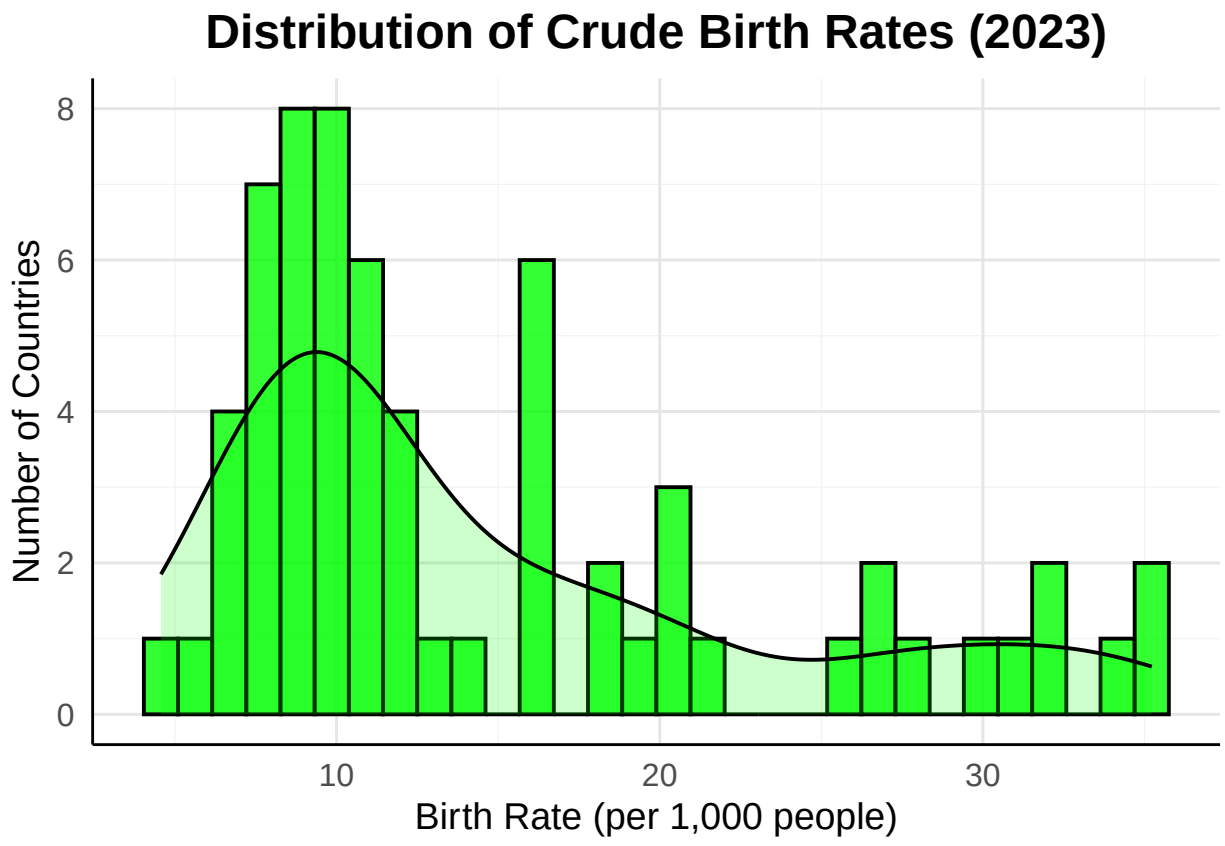
Birth Rate Distribution (Dependent Variable)

```r
ggplot(data_2023_clean, aes(x = birth_rate)) +
  geom_histogram(bins = 30, fill = "green", alpha = 0.8, color = "black") +
  geom_density(aes(y = ..count..), fill = "green", alpha = 0.2) +  # Adding density overlay
  labs(
    title = "Distribution of Crude Birth Rates (2023)",
    x = "Birth Rate (per 1,000 people)",
    y = "Number of Countries"
  ) +
  theme_minimal(base_size = 15) +  # Increase base size for readability
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    panel.grid.major = element_line(color = "gray90", size = 0.5),
    panel.grid.minor = element_line(color = "gray95", size = 0.2),
    axis.line = element_line(color = "black", size = 0.5)
  )
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



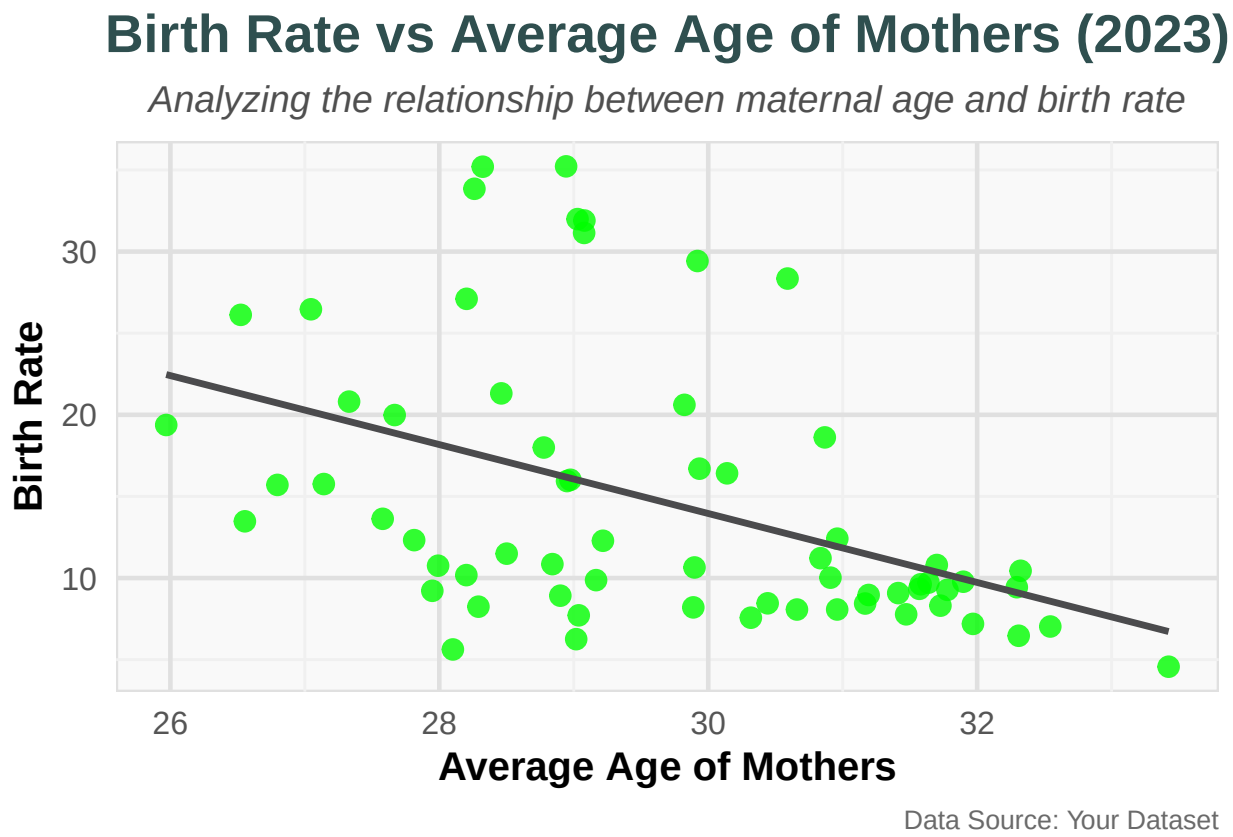
Birth Rate vs Average Age of Mothers

```
ggplot(data_2023_clean, aes(x = avg_age_mothers, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with a slight transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker black line for the
  labs(
    title = "Birth Rate vs Average Age of Mothers (2023)",
    subtitle = "Analyzing the relationship between maternal age and birth rate",
    x = "Average Age of Mothers",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
```

```
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

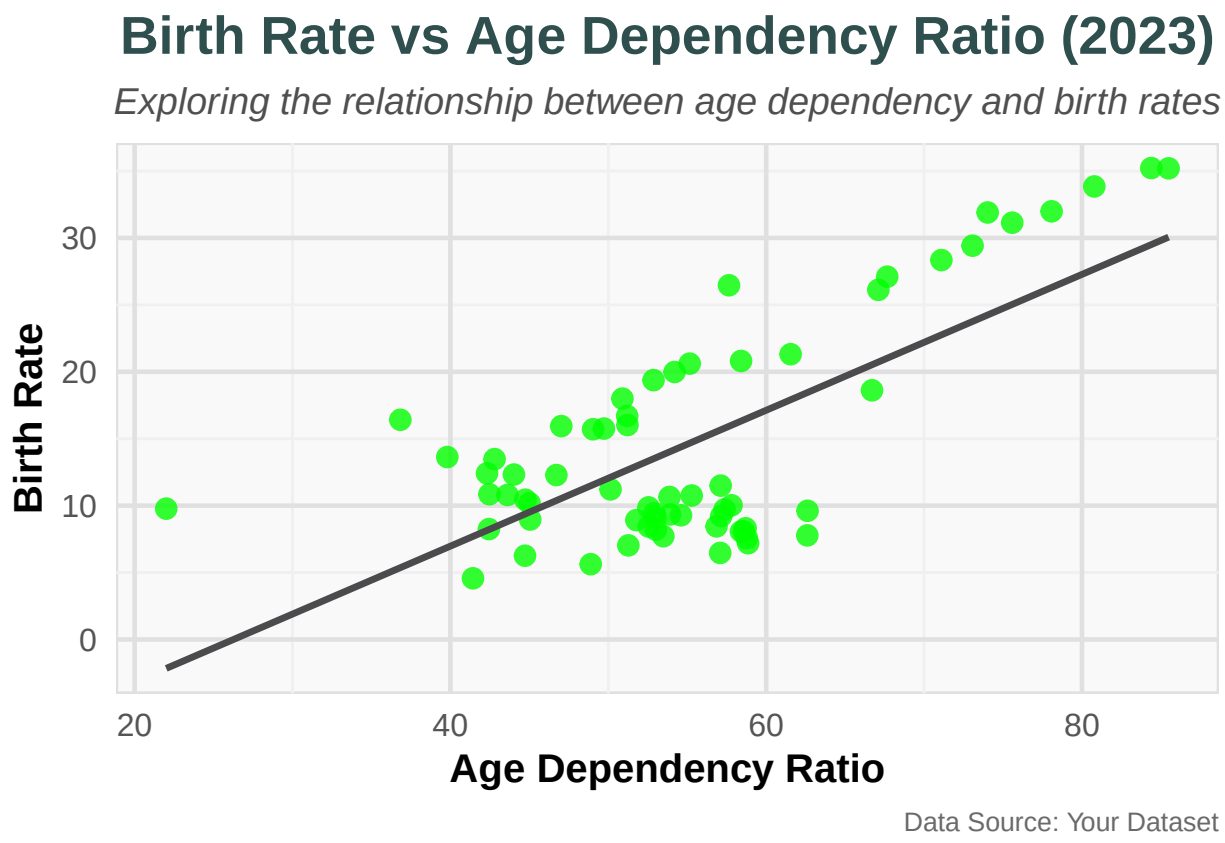
```
## `geom_smooth()` using formula = 'y ~ x'
```



Birth Rate vs Age Dependency Ratio

```r
ggplot(data_2023_clean, aes(x = age_dependency_ratio, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker regression line
  labs(
    title = "Birth Rate vs Age Dependency Ratio (2023)",
    subtitle = "Exploring the relationship between age dependency and birth rates",
    x = "Age Dependency Ratio",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

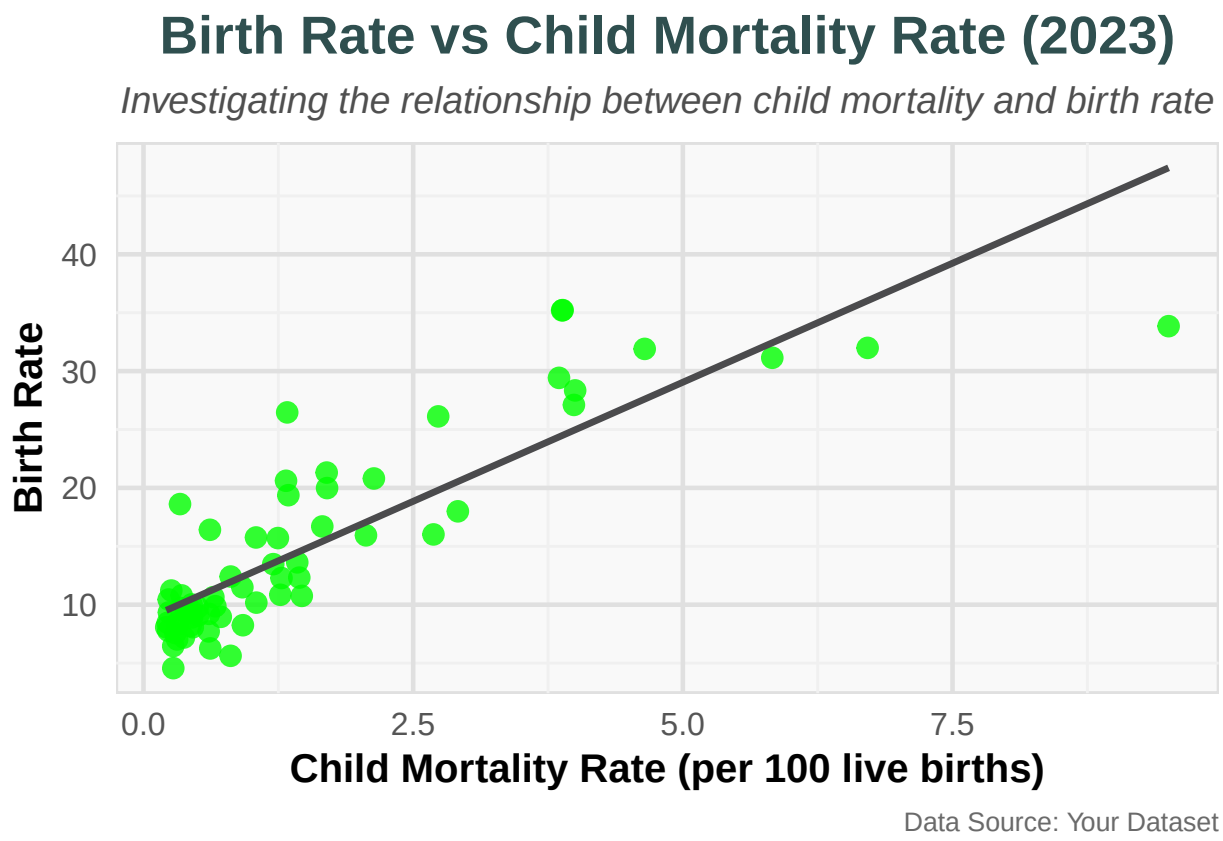
```
## `geom_smooth()` using formula = 'y ~ x'
```



Birth Rate vs Child Mortality Rate

```
ggplot(data_2023_clean, aes(x = child_mortality_rate, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker regression line
  labs(
    title = "Birth Rate vs Child Mortality Rate (2023)",
    subtitle = "Investigating the relationship between child mortality and birth rate",
    x = "Child Mortality Rate (per 100 live births)",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

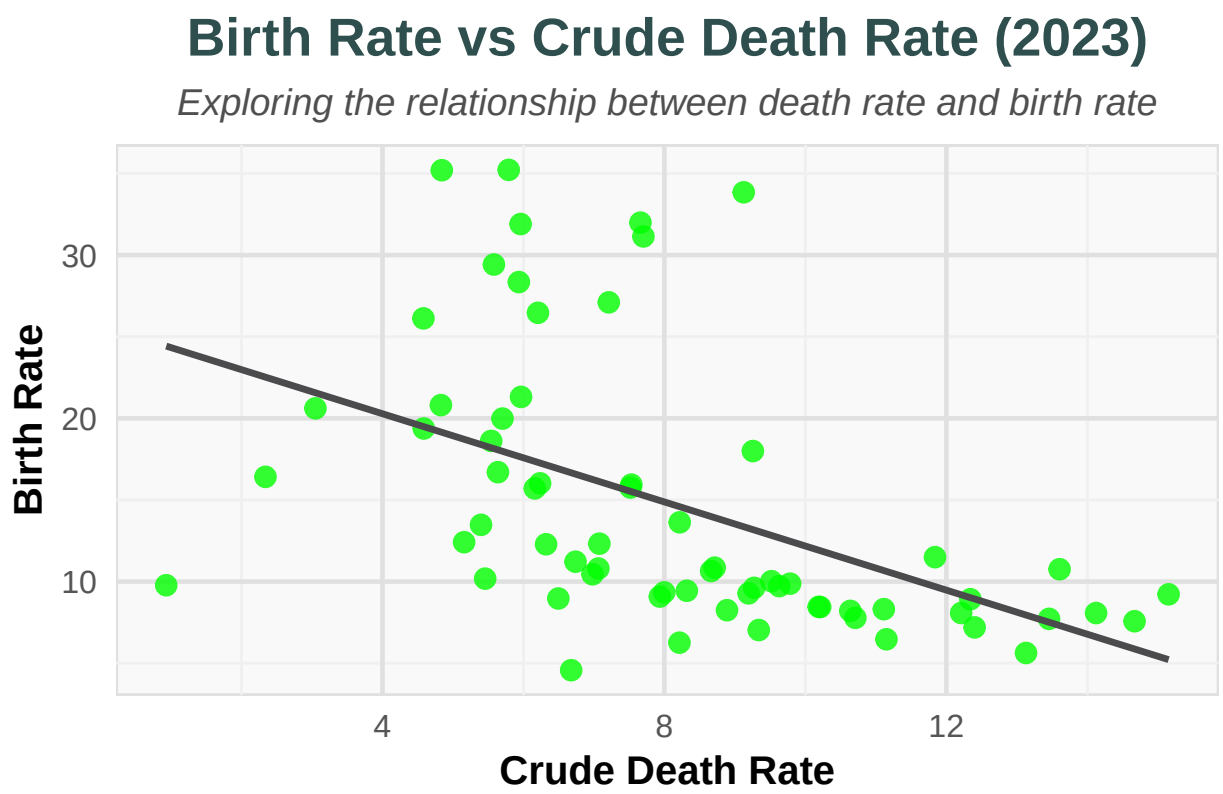
```
## `geom_smooth()` using formula = 'y ~ x'
```



Birth Rate vs Crude Death Rate

```
ggplot(data_2023_clean, aes(x = crude_death_rate, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker regression line
  labs(
    title = "Birth Rate vs Crude Death Rate (2023)",
    subtitle = "Exploring the relationship between death rate and birth rate",
    x = "Crude Death Rate",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

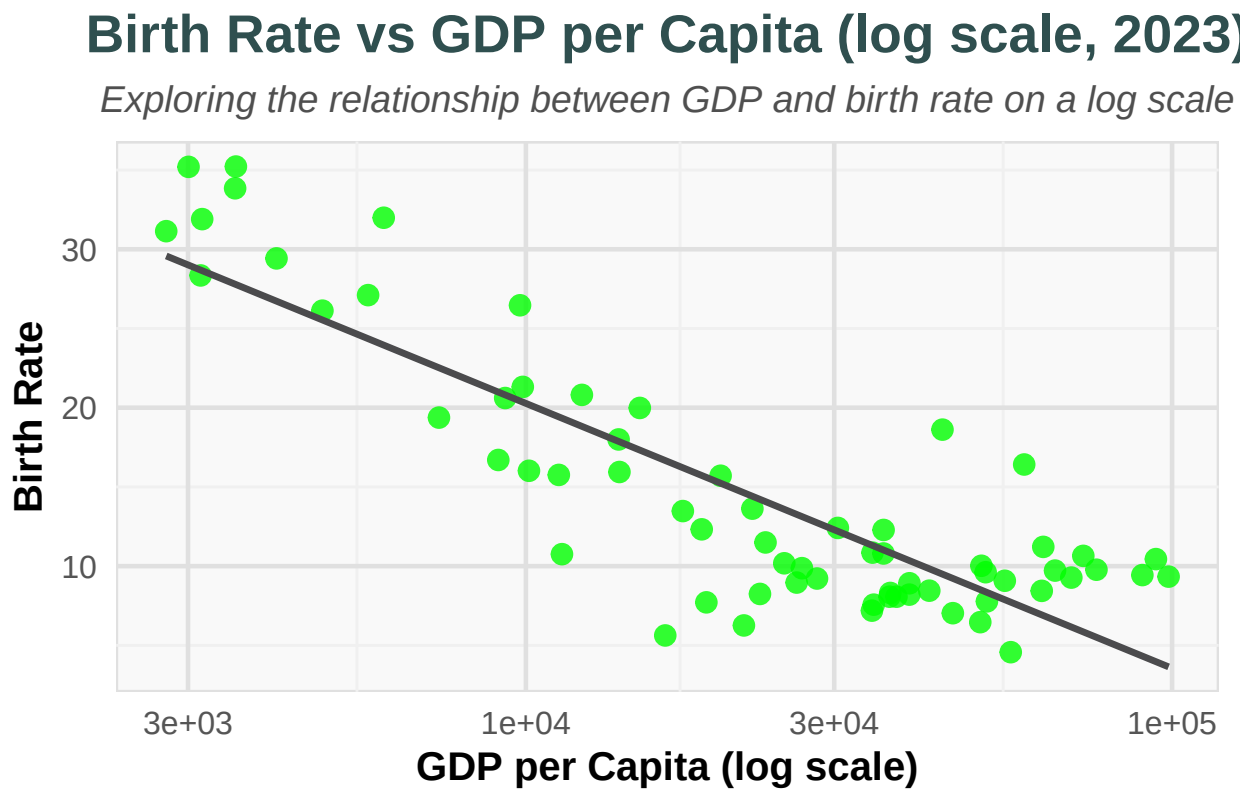


Birth Rate vs GDP per Capita (Log Scale)

```
ggplot(data_2023_clean, aes(x = gdp_per_capita, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker regression line
  scale_x_log10() +  # Log scale for x-axis
  labs(
    title = "Birth Rate vs GDP per Capita (log scale, 2023)",
    subtitle = "Exploring the relationship between GDP and birth rate on a log scale",
    x = "GDP per Capita (log scale)",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
```

```
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

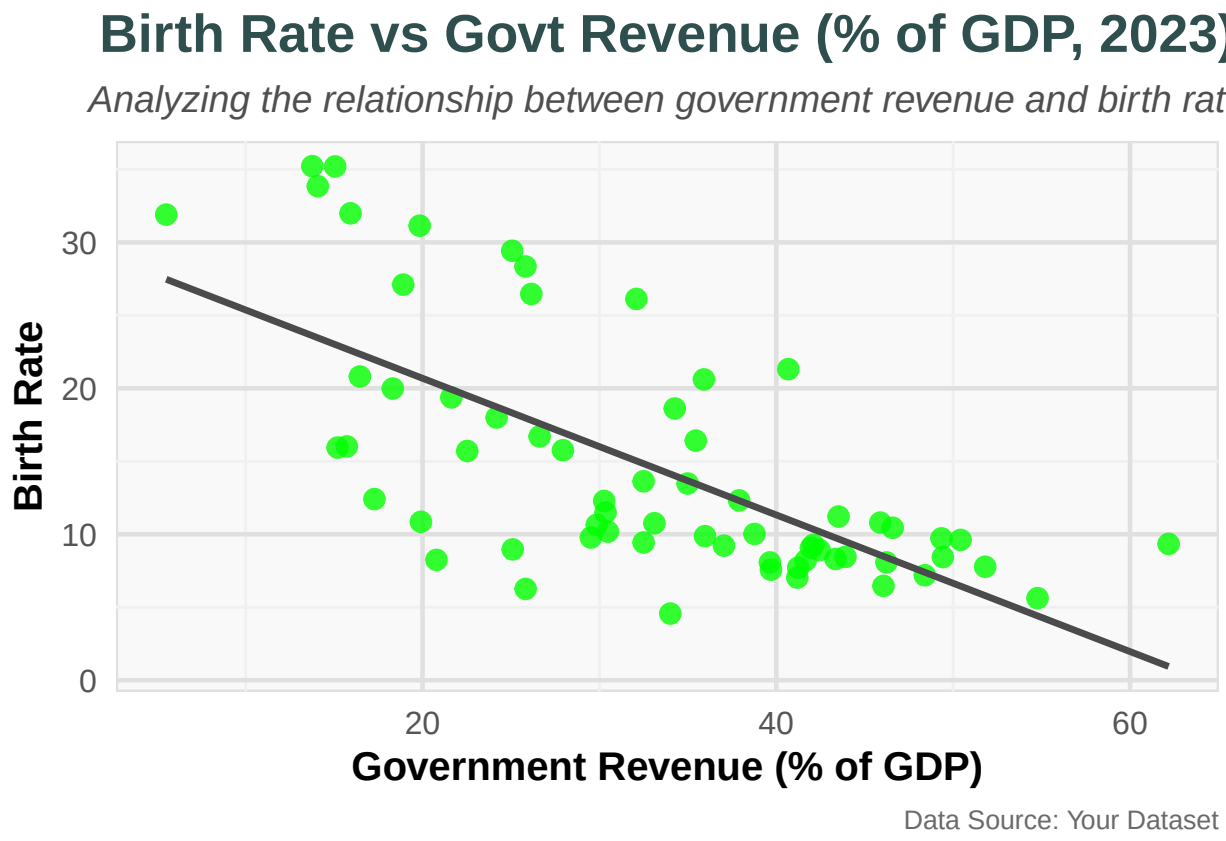


Birth Rate vs Government Revenue (% of GDP)

```
ggplot(data_2023_clean, aes(x = gov_revenue_gdp, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker regression line
  labs(
    title = "Birth Rate vs Govt Revenue (% of GDP, 2023)",
    subtitle = "Analyzing the relationship between government revenue and birth rate",
    x = "Government Revenue (% of GDP)",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
```

```
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

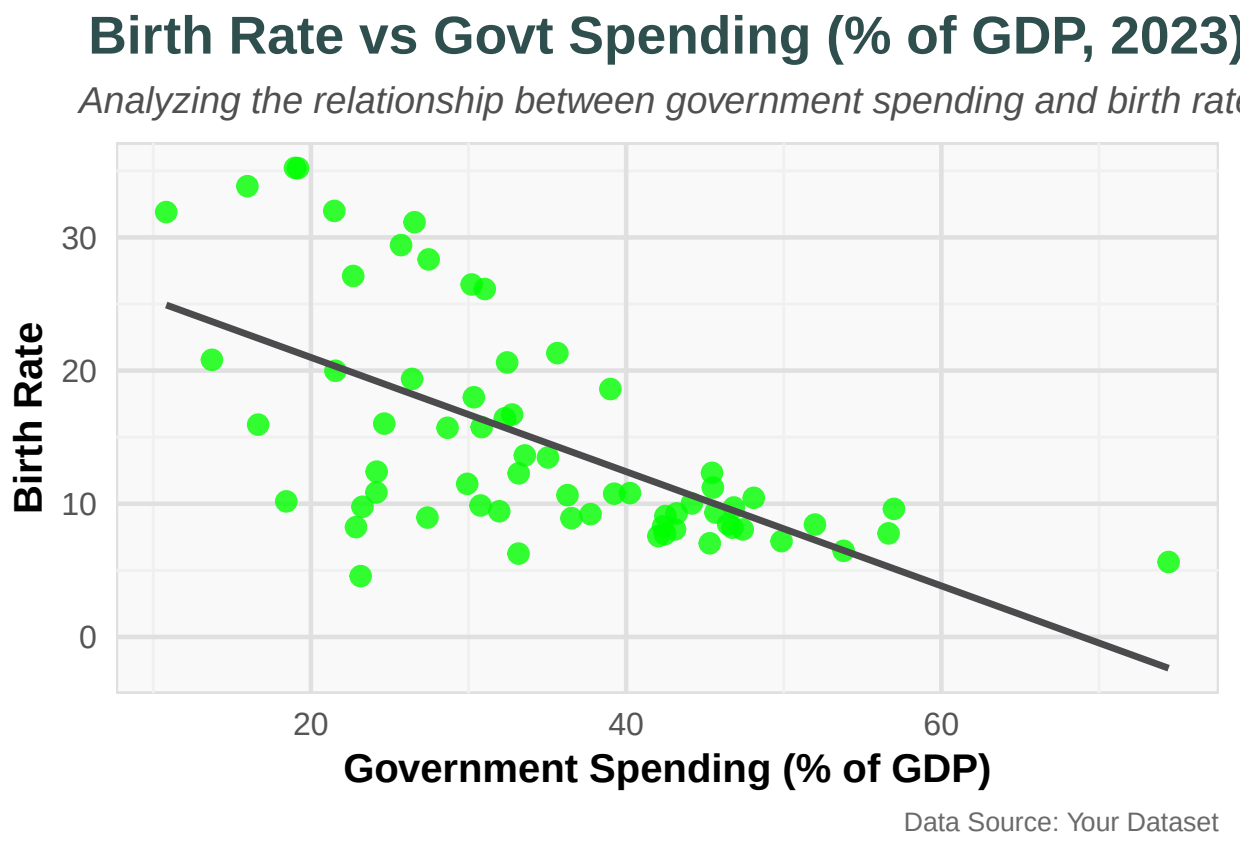


Birth Rate vs Government Spending (% of GDP)

```
ggplot(data_2023_clean, aes(x = gov_spending_gdp, y = birth_rate)) +
  geom_point(alpha = 0.8, color = "green", size = 3) +  # Dark green with transparency
  geom_smooth(method = "lm", se = FALSE, color = "#4B4B4D", size = 1.2) +  # Thicker regression line
  labs(
    title = "Birth Rate vs Govt Spending (% of GDP, 2023)",
    subtitle = "Analyzing the relationship between government spending and birth rate",
    x = "Government Spending (% of GDP)",
    y = "Birth Rate",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 16) +
```

```
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color = "#2F4F4F"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "#4F4F4F"),
    plot.caption = element_text(hjust = 1, size = 10, color = "#6A6A6A"),
    axis.title = element_text(size = 15, face = "bold"),
    axis.text = element_text(size = 12, color = "#555555"),
    panel.grid.major = element_line(color = "#E0E0E0", size = 0.8),  # Light grey grid lines
    panel.grid.minor = element_line(color = "#F0F0F0", size = 0.5),  # Lighter minor grid lines
    panel.background = element_rect(fill = "#F9F9F9", color = "#E0E0E0"),  # Soft background
    plot.background = element_rect(fill = "#FFFFFF")  # White background for the plot
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Correlation Heatmap

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.5.2
```

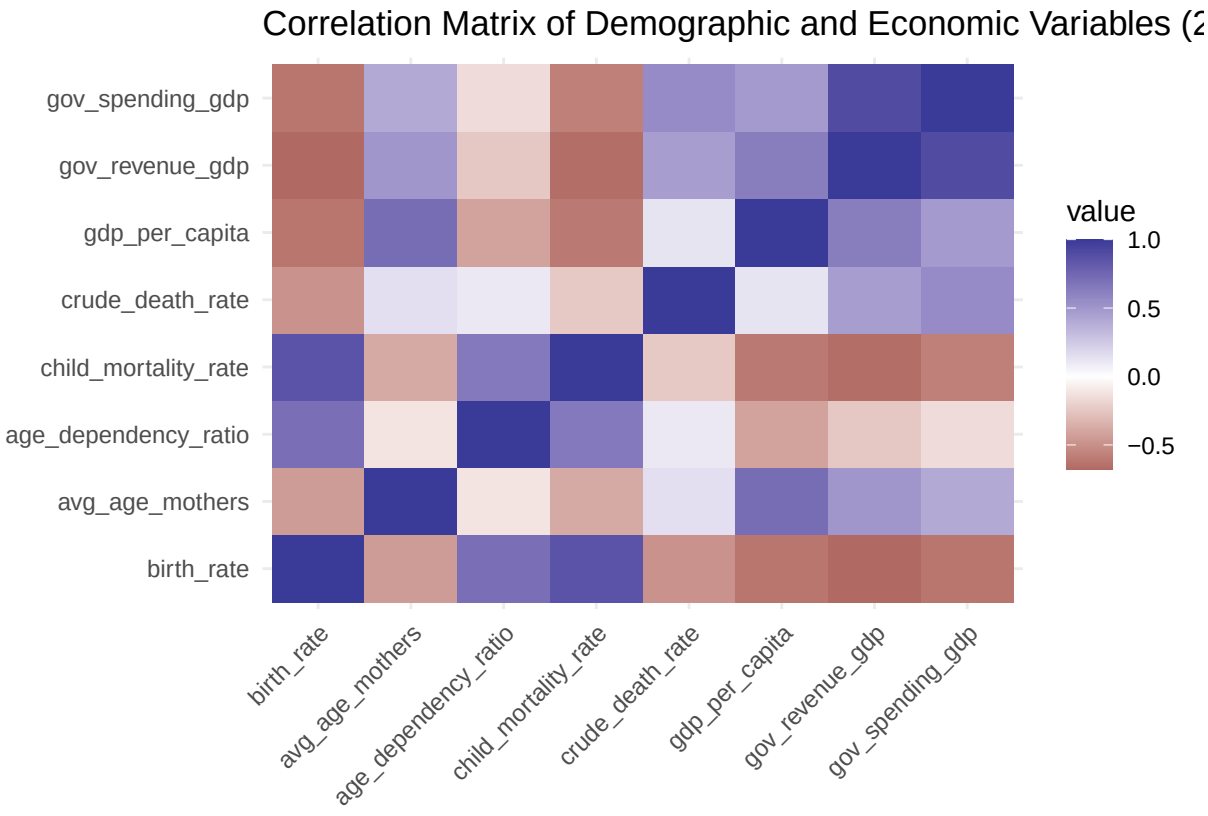
```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
corr_data <- data_2023_clean %>%
  select(
    birth_rate,
    avg_age_mothers,
    age_dependency_ratio,
    child_mortality_rate,
    crude_death_rate,
    gdp_per_capita,
    gov_revenue_gdp,
    gov_spending_gdp
  )

corr_matrix <- cor(corr_data)
corr_melt <- melt(corr_matrix)

ggplot(corr_melt, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(midpoint = 0) +
  labs(
    title = "Correlation Matrix of Demographic and Economic Variables (2023)",
    x = "",
    y = ""
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Correlation Matrix of Demographic and Economic Variables (2
gov_spending_gdp
gov_revenue_gdp
gdp_per_capita
crude_death_rate
child_mortality_rate
age_dependency_ratio
avg_age_mothers
birth_rate
birth_rate
avg_age_mothers
age_dependency_ratio
child_mortality_rate
crude_death_rate
gdp_per_capita
gov_revenue_gdp
gov_spending_gdp
value
1.0
0.5
0.0
−0.5